2006

Design of linear Boolean network codes for combination networks

Shoupei Li
Louisiana State University and Agricultural and Mechanical College, sli2@lsu.edu

DESIGN OF LINEAR BOOLEAN NETWORK CODES FOR COMBINATION NETWORKS

A Thesis

Submitted to the Graduate Faculty of the
Louisiana State University
and Agriculture and Mechanical College
in partial fulfillment of the
requirements for the degree of
Master of Science in Electrical Engineering

in

The Department of Electrical & Computer Engineering

by
Shoupei Li
B.S. in Mathematics, Fudan University, 2001
M.S. in Mathematics, Fudan University, 2004
May, 2006

Acknowledgements

First and foremost I would like to express my sincere appreciation and thanks to Prof. Xue-Bin Liang for his constant guidance and valuable comments throughout my thesis research. I am also very grateful to Prof. Guoxiang Gu and Prof. Morteza Naraghi-Pour for serving as the members of my thesis committee.

I would like to thank my friends at LSU, especially Amariucai George Traian, who has helped a lot in my studies.

Finally, I want to thank my beloved wife, Yan Jiang, for her help and encouragement during these two difficult years. In addition, I would like to thank God for the help, love and support of brothers and sisters in Chinese Christian Church at Baton Rouge.

Table of Contents

List of Figures

Abstract

In the thesis, we investigate linear Boolean network codes on a special class of multicast networks, called combination networks. Using companion matrices of primitive polynomials over finite fields, we design a class of symmetric linear Boolean network codes from Reed-Solomon codes for single-source combination networks. We also prove that, for some cases, the linear Boolean network codes are optimal in the sense of minimum network uses. In the thesis, we further consider two-source network coding problem for combination networks and other specific networks. We develop a method to evaluate an outer bound for these two-source networks. By designing linear Boolean network codes which achieve extreme points of rate regions, we show that the outer bound is actually tight for some of these networks.

Chapter 1

Introduction

Recently lots of researchers are focusing on network coding, which has potential applications in computer networks and wireless communication networks. The concept of network coding is first presented in the simple and beautiful paper [1]. Since then, researchers from different disciplines, such as mathematics, computer science and communication science, rush into this newly born field.

The idea of network coding is to allow information mixing at nodes during information transmission in communication networks. Contrary to our intuition, the throughput of the networks can be significantly increased by mixing information flows at nodes and recovering them at sinks. This is so-called *network coding*.

Unlike physical commodities, information can be added up (or mixed) without increasing the total size. For example, let $b_1, b_2 \in GF(2)$. Using the addition in Galois Field, we add bit b_1 to bit b_2, which yield $b = b_1 \bigoplus b_2$, still one bit. The resulting bit b contains both information of bit b_1 and b_2. Given

additional information, say b_1, we can recover both b_1 and b_2 from b through the equation $b_2 = b \bigoplus b_1$.

The basic fact above explains why information flows behave so different from commodity flows in networks. Before we explore the laws governing information flows in networks, we will first introduce the network model and commodity flows in networks.

1.1 Commodity Flows in Networks

Many communication networks can be modeled as directed graph $G = (V, E)$, where V is the set of nodes, E is the set of edges. For each edge $(i, j) \in E$, we assign a nonnegative number, R_{ij}, called the capacity of the edge. In V, there is a source node, s, where commodity is sent. And there is a destination, t, where commodity is required. We call this node a sink.

Let $i_1, i_2, \cdots, i_n (n \geq 2)$ be a sequence of distinct nodes in V, such that $(i_k, i_{k+1}) \in E$, for $k = 1, 2, \cdots, n-1$. Then the sequence

$$i_1, (i_1, i_2), i_2, \cdots, (i_{n-1}, i_n), i_n$$

is called a chain from i_1 to i_n. If i_1 and i_n represent the same node, then the chain is called a cycle in G. A directed graph G is called acyclic if it does not contain any cycle, otherwise it is called cyclic.

Let $U \subseteq V, \overline{U} = V \setminus U$, if $s \in U$, and $t \in \overline{U}$, then U is called a cut separating s from t. The capacity of cut U is defined as

$$C(U) = \sum_{i \in U, j \in \overline{U}} R_{ij}$$

Since we only consider finite directed graphs, the number of cuts separating s from t is also finite. Therefore, we can find a cut whose capacity is minimal among all the cuts. This cut is called a minimal cut. Usually the minimal cut is not unique in a directed graph.

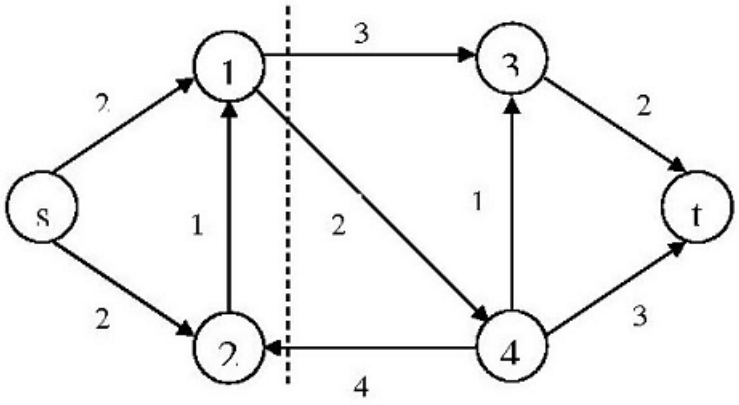

Figure 1.1: A Directed Graph

Let us consider the directed graph in Figure 1.1, $U = \{s, 1, 2\}$ is a cut separating s from t, its capacity is

$$C(U) = R_{1,3} + R_{1,4} = 5$$

Obviously, U is not a minimal cut because the capacity of cut $\{s, 2\}$ is 3, less than the capacity of cut U.

Let $\Gamma_+(i) = \{j : (j, i) \in E\}, \Gamma_-(i) = \{j : (i, j) \in E\}$. A commodity flow in network G is a function f defined as

$$f : E \to R^+$$

where R^+ is the set of non-negative real numbers, and f satisfies two condi-

tions:

$$\sum_{j\in\Gamma_-(i)} f(i,j) - \sum_{j\in\Gamma_+(i)} f(j,i) = \begin{cases} v & \text{if } i = s; \\ 0 & \text{if } i \neq s,t; \\ -v & \text{if } i = t. \end{cases} \tag{1.1}$$

$$f(i,j) \leq R_{ij}, \text{for all } (i,j) \in E \tag{1.2}$$

Condition (1.1) can be regarded as conservation law of commodity flow. For any intermediate node, the amount of commodities flowing out of the node is equal to the amount of commodities flowing into this node. Condition (1.2) is the capacity constraint of the flow along each edge. It states that the amount of flow along the edge can not exceed the capacity of this edge.

In reality, gas (or water) pipelines and transportation can be modeled as commodity flows in networks. A natural question regarding to commodity flows arises, that is, what is the maximal flow which satisfies condition (1.1) and (1.2), given a directed network. In other words, what is the maximum value of v if the network topology and edge capacity $\vec{R} = [R_{ij}, (i,j) \in E]$ are known? The following theorem answers the question.

Theorem 1.1.1 (Max-flow min-cut theorem) *Given a directed network* $G = (V, E)$ *and its edge capacity* $\vec{R} = [R_{ij}, (i,j) \in E]$*, the maximal commodity flow from* s *to* t *is equal to the capacity of the minimal cut separating* s *from* t*.*

This milestone result is due to L. R. Ford and D. R. Fulkerson. For more details about commodity flows in networks, readers can see [2]. Figure 1.2 is a maximum flow for the directed graph in the above.

As we have mentioned, information flows behave differently from com-

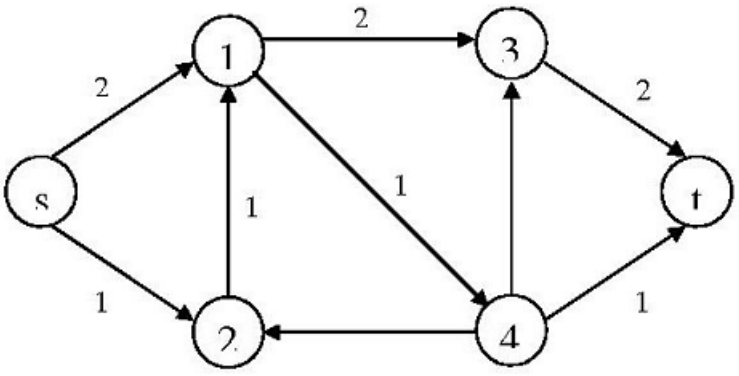

Figure 1.2: A Maximal flow for the Directed Graph

modity flows in networks. In next section, we will explore the laws which are governing the information flows in networks.

1.2 Single-source Information Flows

In communication network $G = (V, E)$, information source X is generated in the source node, s, and then multicast to sinks $t_1, t_2, \cdots, t_L$. Each edge (i, j) models a communication channel with channel capacity R_{ij}. Information is sent from node i to node j without error. At each node, there will be multiple incoming channels. Each node has the ability to copy and encode information received and send it to outgoing channels. Each sink can recover the information source X totally from the information received. This is the Single-source Multicast network model.

Like commodity flows in networks, there exist some laws which govern the information flows, as are shown in the following:

(1) The content of information flowing out of a node is contained in the information received by this node. This is similar to the conversation law of

commodity flows;

(2) The rate of information flow transmitted through an edge is less than the channel capacity of the edge.

Let h denote the rate of information source X. The following theorem characterizes the achievable rate of information source in a single-source multicast networks.

Theorem 1.2.1 [1] *Let $G = (V, E)$ be the single-source multicast network, then*

$$h \leq \min_{\ell=1,\cdots,L} maxflow(s \rightarrow t_\ell)$$

where $maxflow(s \rightarrow t_\ell)$ is the value of maximal commodity flow from s to t_ℓ.

We call $\min_{\ell=1,\cdots,L} \text{maxflow}(s \rightarrow t_\ell)$ the maximal flow bound of the single-source multicast network. The maximal flow bound is tight for single-source multicast networks.

More details about the setup and proof of Theorem 1.2.1 can be seen in [1]. Here we outline the main idea of the proof and give some explanation to the theorem.

Let U be a cut separating s from t_ℓ for some $\ell \in \{1, 2, \cdots, L\}$, and $(U, \overline{U}) = \{(i, j), i \in U, j \in \overline{U}\}$. For any information flow in the network G, sink t_ℓ has to recover information source X from all information received from its incoming edges. All information sent from source node s to t_ℓ must pass through the edges in $(U, \overline{U})$. Therefore information source X must be a function of all information flowing through the cut U. Hence the achievable rate of information flows is bounded by the capacity of the cut separating s from t_ℓ. This holds for any sink and any cut.

Given $h \leq \min_{\ell=1,\cdots,L} \text{maxflow}(s \to t_\ell)$, it is shown that there exists at least an information flow which achieves this rate using random procedure.

The information source X can be modeled as a stationary discrete time random process

$$X = (X_1, X_2, \cdots, X_n, \cdots)$$

Then, $h = H(X) = \lim_{n\to\infty} H(X_n|X_1, X_2, \cdots, X_{n-1})$. From source coding theorem, when n is large enough, the information source can be compressed into

$$\Omega = \{1, 2, \cdots, \lceil 2^{nh} \rceil\}$$

where $\lceil x \rceil$ denotes the largest integer less than x. And each symbol in Ω is selected with uniform distribution, i.e., each symbol in Ω will appear with probability $|\Omega|^{-1}$.

Nodes in acyclic directed networks can be sorted in the following way. $(i, j) \in E$ implies $i < j$. Then we can construct random information flows in the acyclic directed network.

First expand Ω into $\Omega' = \{1, 2, \cdots, \lceil C2^{nh} \rceil\}$,with $C > 1$. We define a flow f as follows. For any $x \in \Omega'$, $f_{ij}(x)$ chooses a symbol from $\{1, 2, \cdots, \eta_{ij}\}$ uniformly. η_{ij} satisfies

$$\log_2 \eta_{ij} \leq R_{ij} + \epsilon$$

Let g_{t_ℓ} be the decoding function of sink t_ℓ, $g_{t_\ell}(x) \in \Omega'$. A symbol x can be distinguished by sink t_ℓ if and only if $g_{t_\ell}(x) \neq g_{t_\ell}(x')$, for all other $x' \in \Omega'$. A symbol is said to be distinguished if it can be distinguished by all sinks.

Then the average number of symbols which can be distinguished in Ω' is approximately

$$(1-\delta(n,L))C2^{nh}$$

where $\delta(n,L)$ tends to zero as $n \to \infty$. Therefore there exists at least on information flow in these randomly constructed flows, which can distinguish at least $\lceil 2^{nh} \rceil$ symbols. We pick up all such symbols and form a new set, still denoted by Ω. Then the rate h is achievable by one such information flow defined on the set Ω.

For a cyclic directed network, we can transform it into an acyclic directed network and apply the same technique to construct network codes.

When $L = 1$,that is, there is only one sink in the network, network coding does not increase the throughput of the network. Information flows behave the same as commodity flows in such networks. when $L \geq 2$, the advantage of network coding can be seen in [3]. Especially in [5], Prof. Xue-Bin Liang investigates the throughput gap between network switching and network coding for single-source multicast networks, and finds that the gap is upper bounded by nth *harmonic number*. And this harmonic-number bound is *asymptotically tight* for combination networks.

So far, we know that there exist network codes which achieve the maxflow bound, but [1] does not provide an explicit method to construct such network codes.

In [4], a linear algebraic structure, called *linear-code multicast*(LCM), is developed to achieve the maxflow bound. In [4], Communication networks are modeled as directed graph with unit capacity edges, i.e., the capacity of

each edge is one symbol per unit time.

An LCM on a communication network is an assignment of h-dimensional vector over finite field $GF(q)$ to each edge. For each node, there is a vector space linearly spanned by vectors assigned to its incoming edges. The vectors assigned to its outgoing edges are selected from the vector space.

If the vectors assigned to edges are made as independent as possible, then it is more possible for sinks to receive a full rank matrix so that they can recover the information source. A more specific structure of LCM, called generic LCM, is developed in [4]. Generic LCM can guarantee that those vectors assigned to edges are largely independent. In [4], the important result is given,

Theorem 1.2.2 *Generic LCM can achieve the maxflow bound. And generic LCMs exist for any acyclic communication network, given that the base field is large enough.*

However [4] does not provide explicit method to construct generic LCM. Following this line, [3] designs a polynomial time algorithm to construct generic LCMs for acyclic communication networks. Let T be the set of sinks, then the network can be decomposed into $|T|$ sub networks. Each sub network has single source and single sink, and there are h distinct paths from the source to the sink. Along these h distinct paths, h-dimensional vectors over finite field with size larger than $|T|$ are assigned, such that, any h vectors on different paths are linear independent. One edge may be shared by more than one sink. These edges sometimes could be the "bottleneck" of information flows. They are the exact points where flows need to be mixed.

On these edges, we assign vectors which maintain the independence within several sub networks. The core technique in the algorithm developed in [3] is to construct such vectors which maintain independence among several groups of vectors. If a node only receives an incoming vector, then it can simply relay the vector to its successors.

T.Ho *et al.* [8] analyzed the performance of randomized linear codes based on the algebraic framework developed in [9], and show that the chance for all sinks to receive full rank matrices is very high provided the base field is large enough.

Theorem 1.2.3 *For a linear randomized network code in which all code coefficients are chosen independently and uniformly in a finite field F_q, the probability that all receivers can receive full source information is at least*

$$(1 - \frac{|T|}{q})^h$$

where T is the set of receivers and h is the maxflow bound.

The randomized approach does not need central knowledge of network topology. When the size of the networks is huge (e.g., world wide web), or the network is dynamic, it is almost impossible to know the topology of the networks. Randomized linear network code is a good alternative for these networks. Lots of simulations are based on the randomized approach.

So far, the network codes constructed are symbol-level linear codes. Some authors in [10], [11] and [12] investigate the relation between symbol-level linear code and binary linear code. It is shown in [10] that a binary linear code exists as long as a symbol-level linear code exist in a single-source multicast network. However, in [10] and [12], several counter-examples are given to

show that it is not the same case for multi-source networks. In this thesis, we will investigate linear Boolean network codes with minimum network uses on a special class of networks.

1.3 Multi-source Information Flow Problem

Let $X_1, X_2, \cdots, X_N$ be mutually independent information sources. For a communication network $G = (V, E)$, let $S \subseteq V$ be the set of source nodes, and $T \subseteq V$ the set of sinks. Source nodes generate information sources and each sink demands a subset of the information sources. We define two mappings

$$f : \{1, 2, \cdots, N\} \to S$$

and

$$g : \{1, 2, \cdots, N\} \to T$$

Here $f(i)$ denotes the set of source nodes which generates information source X_i, $g(i)$ denotes the set of sinks which demand information of X_i.

For arbitrary mappings of f and g, what is the maximal information rate of $X_1, X_2, \cdots, X_N$, such that all sinks can receive the total information they demand? This is the multi-source problem many researchers are trying to solve. There is only partial answer to this question even with acyclic communication networks. For cyclic networks, the answer is totally unknown.

Among the many literatures on network coding, only a few papers address this hard problem. Let ω_i be the information rate of X_i, our goal in multi-source problem is to characterize the rate region

$$R = \{(\omega_1, \omega_2, \cdots, \omega_N) : \omega_i \text{ can be achieved simultaneously.}\}$$

The best result obtained so far to characterize the achievable rate region R is presented in [13] and [14]. There they gave an inner and outer bound in terms of entropy space to characterize the rate region R.

$$R_{in} \subseteq R \subseteq R_{out}$$

These two bounds can not be evaluated explicitly. So a linear programming bound, called LP bound is developed as an outer bound. Though LP bound can be evaluated, the evaluation is involved. Therefore, [15] develop an outer bound, called *time-sharing bound*, which can be evaluated explicitly. Though the bound is significantly improved from maxflow bound, this bound is not tight in general. Time-Sharing bound can be implied by linear programming bound.

Though multi-source problem is not yet completely solved, some results on special networks have been obtained. In [16] and [17], they independently discovered that the maxflow bound still remains tight for single-source-node two-sink networks.

Is the multi-source problem too general to be a problem, or there does not exist sufficient tool and wisdom to attack the general multi-source problem?

1.4 Applications of Network Coding

With single-source network codes well explored, lots of researchers are dedicated to applications of network coding in wireless ad hoc networks and computer networks.

In [18], a practical network coding scheme based on random codes is developed. Their scheme address the synchronization problem of network coding using buffering technique. In their systems, they address real packet networks in which information is delivered in packets. there are random losses and delays in transmission. And networks will experience links and nodes failure. Their systems approach to the real internet. The networks are huge, to know their graph topology is almost impossible. To counteract these problems, they propose a packet format which includes encoding function along each edge. During network coding, they choose random coefficients in a finite field with size equal to powers of two. The chance for each sink to receive a matrix with full rank is big. The random approach removes the need for central knowledge of graph topology as [3] does. To address the synchronization problem, the label the packets with generation numbers. Nodes only encode the current "generation" packets and discard old packets. They simulated the scheme on graphs of several Internet Service Providers(ISP), and found that their scheme can achieve a rate close to the maxflow bound and the delay of networks is significantly alleviated.

Network coding is very promising in large file content distribution systems, because there is no synchronization problem in these systems. In [19], a new scheme for large scale content distribution based on network coding is proposed. In this scheme, a large file is split into small blocks. These small blocks are treated as symbols in network coding. During the distribution of these blocks, when a node receive several blocks of the file, it then serves as a server, other nearby nodes can download the linear combination of blocks received by this node. A node continues to download blocks from its neighbors

until it receive enough blocks to recover the whole file. Through simulation on practical scenarios, it is demonstrated that the expected download time for the scheme based on network coding improve $20\% \sim 30\%$ compared to coding only at the server, and improve $2 \sim 3$ times compared to sending raw information without coding at all. Furthermore, it is shown that the file distribution system with network coding is more robust to link failures and node departures.

Based on the system present in [19], A real system, called *Avalanche*, has been implemented by Microsoft. This system may outperform the best large file distribution end system, BitTorrent , which is extremely popular as a way of delivering large file.

Though the notion of network coding is inspired from computer networks, network coding may be applied to wireless ad hoc networks. There is a lot of advantage using network coding on wireless networks. To name a few, energy saving and bandwidth saving. For more details of the benefits of network coding for wireless networks, readers can refer to [20]. There a lot of systems are based on randomized network coding.

From the applications of network coding, we can see that randomized network coding is the core attribution to real communication networks. More is to be done to make application to satellite communications, where there are multiple transmitters and multiple receivers.

1.5 Outline of Thesis

The thesis gives a brief review of network coding and develops an approach to construct linear Boolean network codes for a special class of highly symmetric networks, called Combination networks. Through the construction of network codes, we provide an insight into network coding, and advance another hard issue in network coding, —optimization of network codes. In our thesis, we prove that the network codes we constructed are optimal in some cases. Also we want to shed some light on multiple-source problem through some examples.

In Chapter 2, we give a detailed description of the construction of linear Boolean network codes for Combination Networks; In Chapter 3, we investigate two examples of multi-source networks and give a uniform method to evaluate a tighter bound for their rate regions. Chapter 4 will give an conclusion and future work we can do on network coding.

Chapter 2

Linear Boolean Network Codes

2.1 Definitions

Definition [21] An $\binom{m}{r}$ combination network is a 3-layer single source multicast network. The first layer consists of the source node, s, where information source is generated. The second layer consists of m intermediate nodes, each of them has an incoming edge from the source node. The third layer consists of $\binom{m}{r}$ nodes, each of them has incoming edges from a unique set of r out of m intermediate nodes.

From the definition above, we can see that combination networks are a class of highly symmetric networks. The codes on these networks also have highly symmetric algebraic structure. Figure 2.1 is a $\binom{3}{2}$ combination network. This network is of special interest in digital information storage.

We restrict that every edge in combination networks has unit capacity, e.g., 1 bit per unit time. We only consider $2 \leq r \leq m-1$. For $r = 1, m-1$, combination networks are degenerated.

It is easy to see that the maxflow bound for $\binom{m}{r}$ combination network is

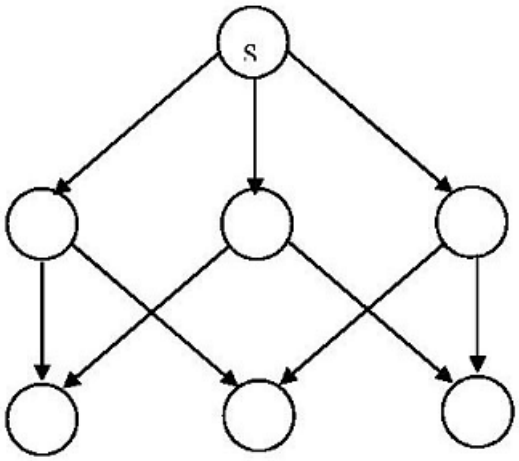

Figure 2.1: $\binom{3}{2}$Combination Network

r. In [3] and [6], it is pointed out that MDS codes (see [7])over a large enough finite field can be used to achieve this bound. In the next few sections, we will use another approach to construct linear Boolean network codes for combination networks. Before we go to the detail, we give two more definitions.

Definition Linear Boolean network code is a kind of network code which only uses a Boolean operation $\bigoplus$ on bits to encode information.

Definition If there exists a network code on Ω for a communication network with maxflow bound equal to h, then the network use of the code is defined as

$$\lceil \log_2(|\Omega|)/h \rceil$$

Definition A minimal network code is a linear Boolean network code on a communication network, which has minimum network uses.

Given an $\binom{m}{r}$ combination network, what is the minimal network code which can achieve the maxflow bound r? Prof. Xue-Bin Liang proposed this problem and showed that it is equivalent to the following packing problem:

For a positive integer n, what is the maximum possible m, such that there exists a sequence of $rn \times n$ matrices$\{A_i\}_{i=1}^{m}$ over $GF(2)$, in which every r of these matrices form an $rn \times rn$ nonsingular matrix?

Furthermore, Prof. Xue-Bin Liang proved the following upper bound

$$m \leq 2^n + r - 1$$

However, the construction issue (i.e., lower bound) of this packing problem has not been addressed. The main result in this chapter is the design of linear Boolean network codes with minimum network uses for combination networks.

2.2 $\binom{m}{2}$ Combination Networks

For $m = 3$, we can design a linear Boolean network code on $\binom{3}{2}$ combination network as shown in Figure 2.2. Here $b_1, b_2 \in GF(2)$. The number of network use for this code is 1. This is an optimal code for this network in the sense of minimal network uses, for we can not expect a network code with network uses less than 1.

For $m = 4$, no matter how we assign the linear combinations of bits on the edges, we can no design a linear Boolean network code with 1 network use to achieve the maxflow bound. However, we can design a linear Boolean network code with two network uses for $\binom{4}{2}$, as shown in Figure 2.4. We

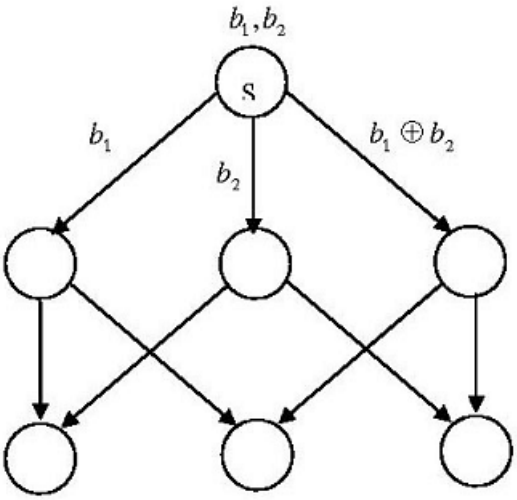

Figure 2.2: Codes on $\binom{3}{2}$Combination Network

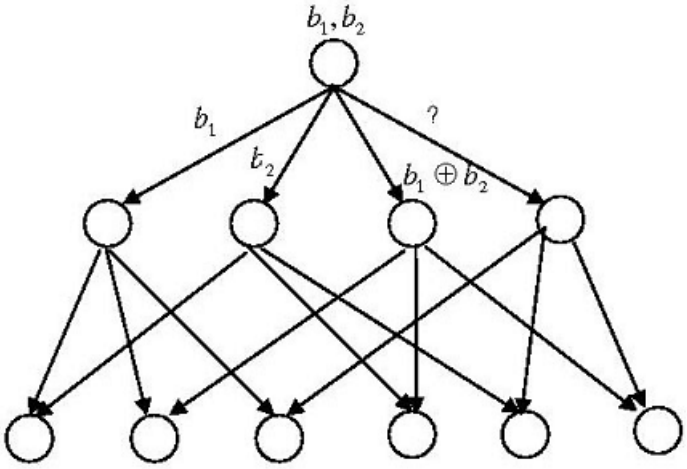

Figure 2.3: $\binom{4}{2}$Combination Network

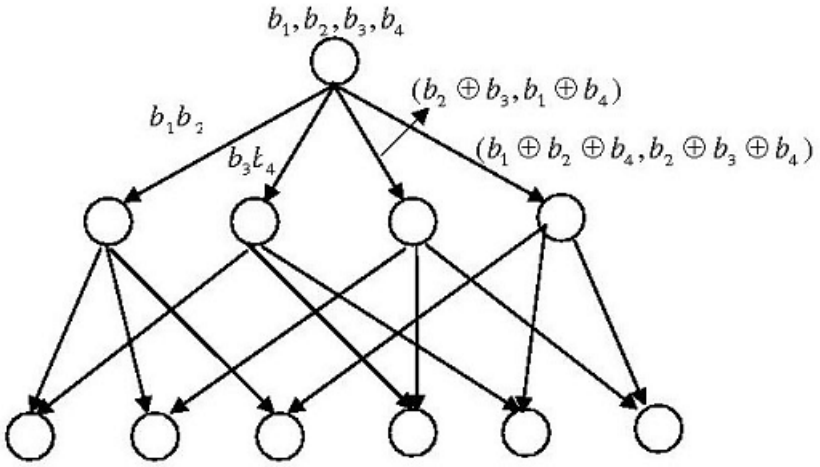

Figure 2.4: Codes on $\binom{4}{2}$ Combination Network

believe that this is the best we can do. A natural question is, given m, what is the minimal network code for $\binom{m}{2}$ combination network? Before we go further, let us analyze the codes in Figure 2.4.

Let $\vec{b} = (b_1, b_2, b_3, b_4)$, we call $\vec{b}$ raw bit vector. We write the codes on edge$(s,1),(s,2),(s,3),(s,4)$ as

$$A_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}, A_2 = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}, A_3 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}, A_4 = \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \\ 1 & 1 \end{pmatrix}$$

We note that $\vec{b} \cdot A_i$ is exactly the code on edge (s,i), for $i = 1,2,3,4$, and every two matrices of $\{A_i\}_{i=1}^4$ form a nonsingular 4×4 matrix. For each sink in $\binom{4}{2}$ combination network receives two matrices in $\{A_i\}_{i=1}^4$, let $\vec{x}$ denotes the information received by this sink, then

$$\vec{x} = \vec{b}(A_i|A_j)$$

for some $i, j \in \{1, 2, 3, 4\}, i \neq j$. Let $A_{ij} \doteq (A_i|A_j)$, A_{ij} is a 4×4 nonsingular matrix. Therefore, we can recover the source information by inverting the matrix A_{ij},

$$\vec{b} = \vec{x}A_{ij}^{-1}$$

As a reminder, all calculations above are in the Galois field $GF(2)$.

For arbitrary $\binom{m}{2}$ combination networks, if we can find m matrices over $GF(2)$, such that every two of them form a nonsingular matrix, then we find a linear Boolean network code for this $\binom{m}{2}$ combination network. Therefore, design of linear Boolean network codes for $\binom{m}{2}$ combination networks is equivalent to the construction of such matrices.

The design of minimal network codes for $\binom{m}{2}$ combination networks can be formulated as a mathematical problem,

Given a positive integer $m > 2$, what is the maximal possible integer of n ,such that there exist a sequence of matrices $A_1, A_2, \cdots, A_m$ over $GF(2)$, satisfying

(1) A_i is a $2n \times n$ matrix, for all $i \in \{1, 2, \cdots, m\}$;

(2) Any two of the matrices $\{A_i\}_{i=1}^m$ form a nonsingular $2n \times 2n$ matrix, i.e.,

$$(A_i, A_j), i, j \in \{1, 2, \cdots, m\}, i \neq j$$

are all nonsingular matrices.

To solve this problem, let us consider its dual problem first,

Given a positive integer n, how large is m, such that there exists a sequence of matrices satisfying condition (1) and (2).

For completeness of this thesis, we give a proof of the following theorem.

Theorem 2.2.1 *For any positive integer n, if there exist a sequence of matrices satisfying condition (1) and (2), then $m \leq 2^n + 1$,and this bound is tight.*

Proof: Suppose $\{A_i\}_{i=1}^m$ satisfy condition (1) and (2). Let $span^*(A_i)$ be the linear span (over $GF(2)$) of columns vectors of A_i, which exclude the zero vector, i.e.,

$$span^*(A_i) = linearspan\{\alpha_{i1}, \alpha_{i2}, \cdots, \alpha_{in}\} \setminus \{0\}$$

where $A_i = (\alpha_{i1}, \alpha_{i2}, \cdots, \alpha_{in})$,$\alpha_{ij}$ is an $2n \times 1$ column vector, for all i, j. We state that

$$span^*(A_i) \bigcap span^*(A_j) = \emptyset$$

for all $i \neq j$.

If $\alpha \in span^*(A_i) \bigcap span^*(A_j)$, then α can be expressed as

$$\begin{aligned} \alpha &= a_1\alpha_{i1} + a_2\alpha_{i2} + \cdots + a_n\alpha_{in} \\ &= b_1\alpha_{j1} + b_2\alpha_{j2} + \cdots + b_n\alpha_{jn} \end{aligned}$$

because $A_{ij} = (A_i|A_j)$ is a $2n \times 2n$ nonsingular matrix, $\{\alpha_{i1}, \alpha_{i2}, \cdots, \alpha_{in}, \alpha_{j1}, \alpha_{j2}, \cdots, \alpha_{jn}\}$ are linear independent. On the other hand,

$$a_1\alpha_{i1} + \cdots + a_n\alpha_{in} - b_1\alpha_{j1} - \cdots - b_n\alpha_{jn} = 0$$

Therefore,$a_i = b_i = 0$, for $i = 1, 2, \cdots, n$. Hence,$span^*(A_i) \bigcap span^*(A_j) = \emptyset$.

Let $E = F_2^{2n} \setminus \{0\}$, where F_2^{2n} is a $2n$-dimensional linear space over $GF(2)$. Then we have

$$\bigcup_{i=1}^{m} span^*(A_i) \subseteq E.$$

It is easy to be seen that

$$|E| = 2^{2n} - 1, |span^*(A_i)| = 2^n - 1$$

for all $i = 1, 2, \cdots, m$. So

$$\begin{aligned} |\bigcup_{i=1}^{m} span^*(A_i)| &= m|span^*(A_1)| \\ &\leq |E| \\ &= 2^{2n} - 1 \end{aligned}$$

Hence,

$$m \leq \frac{2^{2n}-1}{2^n-1} = 2^n + 1$$

To prove the bound is tight, we need to construct $2^n + 1$ matrices which satisfy condition (1) and (2).

Let$\pi(x)$ be a primitive polynomial of $GF(2^n)$ over $GF(2)$, M is the companion matrix of $\pi(x)$. We know that $M^{2^n-1} = I_n$. Let

$$A_1 = \begin{pmatrix} I_n \\ 0 \end{pmatrix}, A_2 = \begin{pmatrix} 0 \\ I_n \end{pmatrix}, A_k = \begin{pmatrix} I_n \\ M^{k-2} \end{pmatrix}, k = 3, \cdots, 2^n + 1$$

It is obvious that these matrices satisfy condition (1). Let

$$A_{ij} = (A_i|A_j)$$

For $j > i \geq 3$,

$$rank(A_{ij}) = rank \begin{pmatrix} I_n & I_n \\ M^{i-2} & M^{j-2} \end{pmatrix}$$

$$
\begin{aligned}
&= rank \begin{pmatrix} I_n & I_n \\ 0 & M^{j-2} - M^{i-2} \end{pmatrix} \\
&= rank \begin{pmatrix} I_n & I_n \\ 0 & M^{i-2}(M^{j-i} - I_n) \end{pmatrix} \\
&= rank \begin{pmatrix} I_n & I_n \\ 0 & M^{j-i} - I_n \end{pmatrix} \\
&= 2n
\end{aligned}
$$

Using similar argument, it is easy to prove that $rank(A_{1j}) = rank(A_{2j}) = 2n, j = 3, \cdots, 2^n + 1$. Therefore, the matrices constructed above are exactly what we want. ∎

Let us turn to our original practical problem. For an $\binom{m}{2}$ combination network, what is the minimum number of network use for a linear Boolean network code to achieve the maxflow bound, and what is the minimal network code?

Theorem 2.2.2 *For an $\binom{m}{2}$ combination network, the minimum number of network uses for a linear Boolean network code to achieve the maxflow bound is $\lceil \log_2(m-1) \rceil$.*

From this theorem, we can see the minimum number of network uses increase as the network size increase. In the proof of Theorem 2.2.1, we provide a method to construct minimal network codes for $\binom{m}{2}$ combination networks.

2.3 $\binom{m}{3}$ Combination Networks

For $m = 4$, we can design a linear Boolean network code with only one network use for $\binom{4}{3}$ combination network, as shown in Figure 2.5. For $m = 5$,

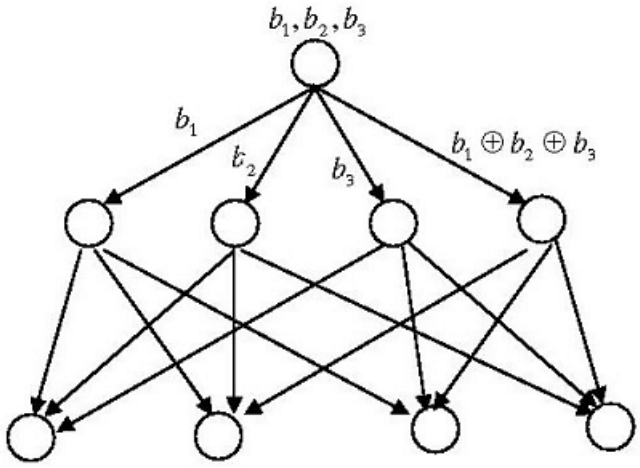

Figure 2.5: Codes on $\binom{4}{3}$ Combination Network

is it possible to design a linear Boolean network code with one network use? The answer is no. But we can design one with two network uses, which is shown in Figure 2.6. Here all $b_i \in GF(2)$.

In $\binom{m}{3}$ combination networks, each sink has three incoming edges. If information flowing along any three of these edges in independent, then every sink can recover the information sent by the source node. Let us analyze the codes in Figure 2.6.

Let $\vec{b} = (b_1, b_2, b_3, b_4, b_5, b_6)$, we call $\vec{b}$ raw bit vector. The codes in Figure 2.6 can be written as matrix form,

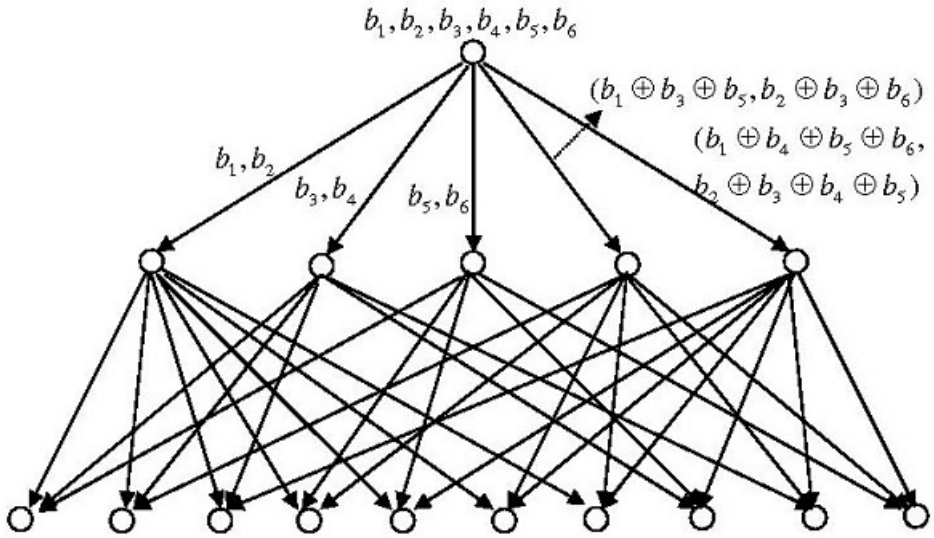

Figure 2.6: Codes on $\binom{5}{3}$Combination Network

$$A_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}, A_2 = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}, A_3 = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}, A_4 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix},$$

$$A_5 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 0 \end{pmatrix}$$

Let $\vec{x}$ denote the information received by some sink, then

$$\vec{x} = \vec{b}(A_i|A_j|A_k)$$

for some distinct $i, j, k \in \{1, 2, 3, 4, 5\}$. Let $A_{ijk} = (A_i|A_j|A_k)$, then A_{ijk} is a 6×6 nonsingular matrix. So the sink can recover the source information by inverting the matrix,

$$\vec{b} = \vec{x}A_{ijk}^{-1}$$

For every sink, they can recover source information by multiplying the the information received by the inverse matrix of their own.

For any $\binom{m}{3}$ combination network, if there exist a sequence of matrices $\{A_i\}_{i=1}^m$ over $GF(2)$,which satisfy the condition that every three of them form a nonsingular matrix, then we can assign these m matrices to the m edges in the first layer. Furthermore, we can multiply these matrices by the raw bit vector, then we obtain a linear binary code for the $\binom{m}{3}$ combination network.

For completeness of this thesis, we give a proof of the following theorem.

Theorem 2.3.1 *For a positive integer n, if there exists a sequence of $3n \times n$ matrices $\{A_i\}_{i=1}^m$ over $GF(2)$, satisfying the condition that, every three of the matrices form a $3n \times 3n$ nonsingular matrix, then*

$$m \leq 2^n + 2$$

and this bound is tight.

Proof: Let $\{A_i\}_{i=1}^m$ be the matrices which satisfy the condition in the theorem, i.e., A_i is a $3n \times n$ matrix over $GF(2)$, and

$$(A_i|A_j|A_k)$$

for distinct i, j, k, are invertible, in other words, the column vectors of $(A_i|A_j|A_k)$ are linear independent. Define

$$span^*(A_i \triangle A_j) = span^*(A_i \bigcup A_j) \setminus (span^*(A_i) \bigcup span^*(A_j))$$

then we state that

$$span^*(A_i \triangle A_j) \bigcap span^*(A_i \triangle A_k) = \emptyset$$

for distinct $i, j, k \in \{1, 2, \cdots, m\}$.

Suppose

$$\alpha \in span^*(A_i \triangle A_j) \bigcap span^*(A_i \triangle A_k).$$

Let

$$A_i = (\alpha_{i1}, \cdots, \alpha_{in}), A_j = (\alpha_{j1}, \cdots, \alpha_{jn}), A_k = (\alpha_{k1}, \cdots, \alpha_{kn}).$$

Then

$$\{\alpha_{i1}, \cdots, \alpha_{in}, \alpha_{j1}, \cdots, \alpha_{jn}, \alpha_{k1}, \cdots, \alpha_{kn}\}$$

are linear independent $3n \times 1$ column vectors. α can be expressed in terms of these vectors,

$$\begin{aligned} \alpha &= a_1\alpha_{i1} + \cdots + a_n\alpha_{in} + b_1\alpha_{j1} + \cdots + b_n\alpha_{jn} \\ &= c_1\alpha_{i1} + \cdots + c_n\alpha_{in} + d_1\alpha_{k1} + \cdots + d_n\alpha_{kn} \end{aligned}$$

where $a_i, b_i, c_i, d_i \in GF(2)$, for all i. Then, we have

$$(a_1 - c_1)\alpha_{i1} + \cdots + (a_n - c_n)\alpha_{in} + b_1\alpha_{j1} + \cdots + b_n\alpha_{jn} - d_1\alpha_{k1} - \cdots - d_n\alpha_{kn} = 0$$

$\{\alpha_{i1}, \cdots, \alpha_{in}, \alpha_{j1}, \cdots, \alpha_{jn}, \alpha_{k1}, \cdots, \alpha_{kn}\}$ are linear independent, therefore,

$$a_1 = c_1, \cdots, a_n = c_n, b_1 = \cdots = b_n = d_1 = \cdots = d_n = 0$$

So $\alpha = a_1\alpha_{i1} + \cdots + a_n\alpha_{in} \in linearspan(A_i)$. But $\alpha \in span^*(A_i \triangle A_j)$, This implies that $\alpha \notin span^*(A_i)$. But $\alpha \neq 0$, hence $span^*(A_i \triangle A_j) \bigcap span^*(A_i \triangle A_k) = \emptyset$.

Let $E = F_2^{3n} \setminus \{0\}$, then

$$(\bigcup_{i=1}^{m} span^* A_i) \bigcup (\bigcup_{j \neq 1} span^*(A_1 \triangle A_j)) \subseteq E$$

$|E| = 2^{3n} - 1$. From the definition and proof above, we know that

$$\begin{aligned} span^*(A_i) \bigcap span^*(A_j) &= \emptyset, \\ span^*(A_i) \bigcap span^*(A_1 \triangle A_j) &= \emptyset, \\ span^*(A_1 \triangle A_j) \bigcap span^*(A_1 \triangle A_k) &= \emptyset \end{aligned}$$

for all distinct i, j, k and $j \neq 1$. On the other hand,

$$|span^*(A_i)| = 2^n - 1$$

and

$$\begin{aligned} |span^*(A_1 \triangle A_j)| &= \left|span^*(A_1 \bigcup A_j)\right| - |span^*(A_1)| - |span^*(A_j)| \\ &= 2^{2n} - 2 \times 2^n + 1 \\ &= (2^n - 1)^2 \end{aligned}$$

Therefore,

$$2^{3n} - 1 = |E|$$

$$
\begin{aligned}
&\geq \left| (\bigcup_{i=1}^{m} span^*A_i) \bigcup (\bigcup_{j\neq 1} span^*(A_1 \triangle A_j)) \right| \\
&= \sum_{i=1}^{m} |span^*(A_i)| + \sum_{j\neq 1} |span^*(A_1 \triangle A_j)| \\
&= m \times (2^n - 1) + (m-1)(2^n-1)^2
\end{aligned}
$$

So, $m \leq 2^n + 2$.

Now, given a positive integer $n \geq 2$, we will construct $2^n + 2$ matrices, every three of which will form a $3n \times 3n$ nonsingular matrix.

Let $\pi(x)$ be a primitive polynomial of $GF(2^n)$ over $GF(2)$, and M be the companion matrix of $\pi(x)$. We set

$$
A_1 = \begin{pmatrix} I_n \\ 0 \\ 0 \end{pmatrix}, A_2 = \begin{pmatrix} 0 \\ I_n \\ 0 \end{pmatrix}, A_3 = \begin{pmatrix} 0 \\ 0 \\ I_n \end{pmatrix}, A_{k+3} = \begin{pmatrix} I_n \\ M^k \\ M^{2k} \end{pmatrix}
$$

for $k = 1, \cdots, 2^n - 1$. All $A_i, (i = 1, \cdots, 2^n + 2)$ are $3n \times n$ matrices.

For any distinct $i, j, k \in \{3, \cdots, 2^n + 2\}$,

$$
\begin{aligned}
rank(A_i|A_j|A_k) &= rank \begin{pmatrix} I_n & I_n & I_n \\ M^i & M^j & M^k \\ M^{2i} & M^{2j} & M^{2k} \end{pmatrix} \\
&= rank \begin{pmatrix} I_n & I_n & I_n \\ 0 & M^j - M^i & M^k - M^i \\ 0 & M^j(M^j - M^i) & M^k(M^k - M^i) \end{pmatrix} \\
&= rank \begin{pmatrix} I_n & 0 & 0 \\ 0 & M^j - M^i & M^k - M^i \\ 0 & M^j(M^j - M^i) & M^k(M^k - M^i) \end{pmatrix}
\end{aligned}
$$

$$
\begin{aligned}
&= rank \begin{pmatrix} I_n & 0 & 0 \\ 0 & I_n & I_n \\ 0 & M^j & M^k \end{pmatrix} \\
&= rank \begin{pmatrix} I_n & 0 & 0 \\ 0 & I_n & 0 \\ 0 & 0 & M^k - M^j \end{pmatrix} \\
&= 3n
\end{aligned}
$$

Using similar argument, it is easy to prove that any three matrices which include A_1, A_2 and A_3, form a nonsingular matrix. So the matrices constructed above is exactly what we want. ∎

Apply the result in the theorem above to the $\binom{m}{3}$ network codes problem, we have

Theorem 2.3.2 *For an $\binom{m}{3}$ combination network, the minimum number of network uses, with which the linear Boolean network code we can construct on this network is*

$$\lceil \log_2(m-2) \rceil.$$

In the proof of theorem 2.3.1, we also give a method to construct the minimal network code for any $\binom{m}{3}$ combination network. Like $\binom{m}{2}$ combination networks, the minimum number of network uses of the linear Boolean network code in $\binom{m}{3}$ network increases as the network size increases.

2.4 $\binom{m}{r}$ Combination Networks($r \geq 4$)

In the previous two sections, we have developed a method to construct minimal network codes for $\binom{m}{2}$ and $\binom{m}{3}$ combination networks, and give a lower bound of network use, with which linear Boolean network codes exist in such networks.

Let us generalize the construction method in the last two sections. Let$\pi(x)$ be a primitive polynomial of $GF(2^n)$ over $GF(2)$, and M the companion matrix of $\pi(x)$. We set

$$A_k = \begin{pmatrix} I_n \\ M^k \\ M^{2k} \\ \vdots \\ M^{(r-1)k} \end{pmatrix} (k = 1, 2, \cdots, 2^n - 1), A_{2^n} = \begin{pmatrix} I_n \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, A_{2^n+1} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ I_n \end{pmatrix}.$$

For $1 \leq i_1 < i_2 < \cdots < i_r \leq 2^n - 1$,

$$rank(A_{i_1}|A_{i_2}|\cdots|A_{i_r}) = rank \begin{pmatrix} I_n & I_n & \cdots & I_n \\ M^{i_1} & M^{i_2} & \cdots & M^{i_r} \\ M^{2i_1} & M^{2i_2} & \cdots & M^{2i_r} \\ \cdots & \cdots & \cdots & \cdots \\ M^{(r-1)i_1} & M^{(r-1)i_2} & \cdots & M^{(r-1)i_r} \end{pmatrix}$$

$$
\begin{aligned}
&= rank \begin{pmatrix} I_n & I_n & \cdots & I_n \\ 0 & M^{i_2} - M^{i_1} & \cdots & M^{i_r} - M^{i_1} \\ 0 & M^{i_2}(M^{i_2} - M^{i_1}) & \cdots & M^{i_r}(M^{i_r} - M^{i_1}) \\ \cdots & \cdots & \cdots & \cdots \\ 0 & M^{(r-2)i_2}(M^{i_2} - M^{i_1}) & \cdots & M^{(r-2)i_r}(M^{i_r} - M^{i_1}) \end{pmatrix} \\
&= rank \begin{pmatrix} I_n & 0 & \cdots & 0 \\ 0 & I_n & \cdots & I_n \\ 0 & M^{i_2} & \cdots & M^{i_r} \\ \cdots & \cdots & \cdots & \cdots \\ 0 & M^{(r-2)i_2} & \cdots & M^{(r-2)i_r} \end{pmatrix} \\
&= n + rank \begin{pmatrix} I_n & \cdots & I_n \\ M^{i_2} & \cdots & M^{i_r} \\ \cdots & \cdots & \cdots \\ M^{(r-2)i_2} & \cdots & M^{(r-2)i_r} \end{pmatrix} \\
&= n + (r-1)n \\
&= rn
\end{aligned}
$$

Similarly, we can check that any r matrices which include one of both of A_{2^n}, A_{2^n+1} form an $rn \times rn$ nonsingular matrix.

Therefore, for an $\binom{m}{r}$ combination network, we can design a linear Boolean network code with $\lceil \log_2(m-1) \rceil$ number of network uses. Compared with the lower bound $\lceil \log_2(m-r+1) \rceil$, the codes constructed above are very close to optimal codes for $4 \leq r < m-1$.

We design an optimal code for $\binom{m}{m-1}$ combination network. Let $b_i \in$

$GF(2), (i = 1, 2, \cdots, m-1)$. We assign $b_1, b_2, \cdots, b_{m-1}$ to the first $m-1$ edges in the first layer, and then we assign $b_1 \bigoplus b_2 \bigoplus \cdots \bigoplus b_{m-1}$ to the last edge in the first layer. Intermediate nodes relay the codes to its outgoing edges. Obviously, the linear Boolean network code achieve the rate $m-1$, while only use the network for one time.

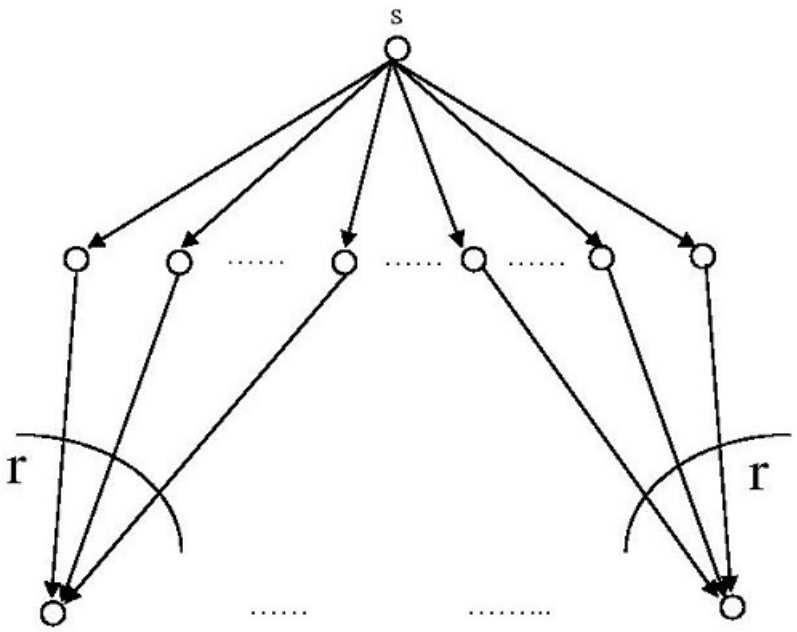

Figure 2.7: General Combination Networks

The relation between the linear Boolean network codes constructed here and linear codes suggested in [6] is that linear Boolean network codes for combination networks can be converted from Reed-Solomon codes using companion matrices. But not all linear Boolean network codes can be converted to Reed-Solomon codes.

Chapter 3

Some Examples of Two-source Networks

3.1 Two-source Networks

Let X, Y be two independent sources generated by one or two source nodes in $G = (V, E)$. Every edge in E has unit capacity, e.g., 1 bit per unit time. For convenience, we use s_1, s_2 represent two source nodes which generate source X and Y respectively. s_1 and s_2 may represent the same node in the graph. Let $S = \{s_1, s_2\}$. And there are multiple sinks $t_1, t_2, \cdots, t_N$. Each sink may demand information from one or both sources. Let T_X denote the set of sinks which demand source X, and T_Y the set of sinks which demand source Y. And $T_X \bigcap T_Y$ may be nonempty. Let ω_1, ω_2 denote the information rate of source X, Y respectively. Then for a two-source network $G = (V, E, T_X, T_Y)$, we want to decide the achievable rate region for the two sources. The maxflow bound for two-source network is an outer bound for

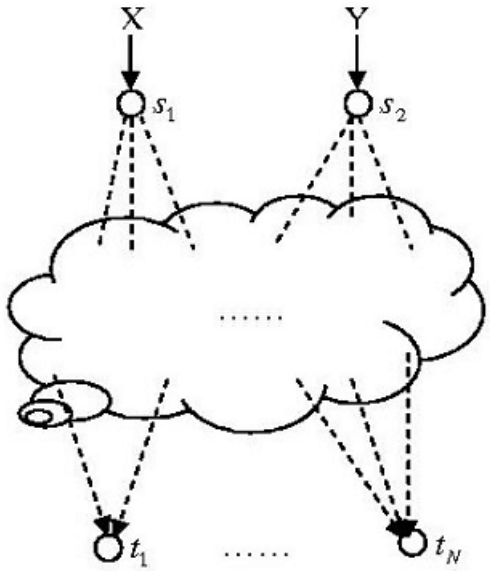

Figure 3.1: Model of Two-source Networks

the achievable rate region. Let

$$\begin{aligned} R_1 &= \bigcap_{T \subseteq T_X} \{\omega_1 \leq mincut(s_1, T)\} \\ R_2 &= \bigcap_{T \subseteq T_Y} \{\omega_2 \leq mincut(s_2, T)\} \\ R_3 &= \bigcap_{T \subseteq (T_X \cup T_Y), T \cap T_X \neq \emptyset, T \cap T_Y \neq \emptyset} \{(\omega_1, \omega_2) : \omega_1 + \omega_2 \leq mincut(S, T)\} \end{aligned}$$

Then the maxflow bound is

$$R = R_1 \bigcap R_2 \bigcap R_3$$

3.2 A Class of Networks Where Maxflow Bound Is Tight

Let us consider a communication network, depicted in Figure 3.2. Source

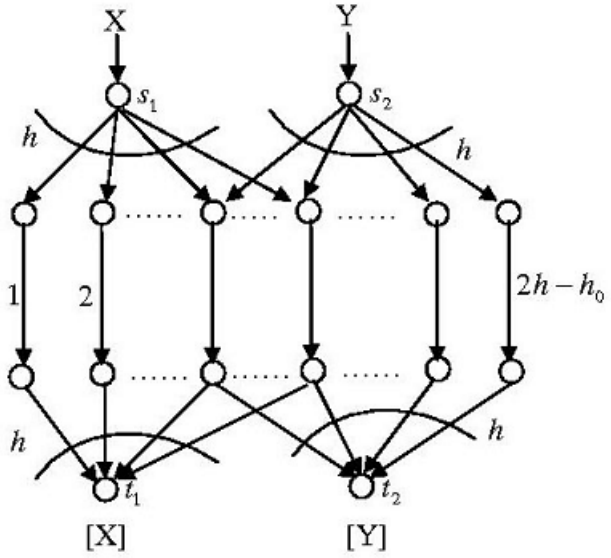

Figure 3.2: A Class of Networks

node s_1 multicast information source X to h nodes in the second layer, s_2 multicast information source Y to h nodes in the second layer. There are totally h_0 nodes in the second layer will receive both X and Y. Then the nodes in the second layer relay information received to the third layer. Sink t_1 has h incoming edges from nodes which receive information X. Similarly, sink t_2 has h incoming edges from nodes which receive information Y. Totally, sink t_1 and t_2 will share h_0 nodes in the third layers. The maxflow bound for this network is

$$R = \{(\omega_1, \omega_2) : 0 \leq \omega_1 \leq h, 0 \leq \omega_2 \leq h, \omega_1 + \omega_2 \leq 2h - h_0\}$$

There are four extreme points in the rate region. They are $\{(0, h), (h - h_0, h), (h, h - h_0), (h, 0)\}$. The rate tuples $\{(0, h), (h, 0)\}$ are easy to achieve, what we need to do is to make one source node "dummy", and treat the network as single source single sink communication network. To show that

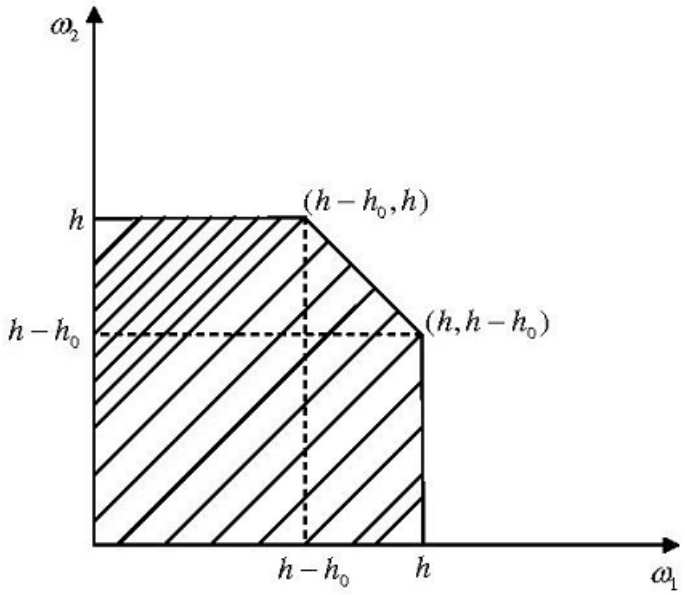

Figure 3.3: Maxflow Bound for These Networks

the rate tuple $(h-h_0, h)$ is achievable, we label the nodes in the second layer as $1, 2, \cdots, 2h-h_0$. Let $a_i \in GF(2), i = 1, 2, \cdots, h-h_0$, and $b_j \in GF(2)$, $j = 1, 2, \cdots, h$. Source node s_1 mulicast $a_i (i = 1, 2, \cdots, h-h_0)$, information bits of X, to the first $h-h_0$ nodes in the second layer. And source node s_2 multicast $b_j (j = 1, 2, \cdots, h)$ to the last h nodes in the second layer. Then the rate tuple $(h-h_0, h)$ is achieved. From symmetry, the rate tuple $(h, h-h_0)$ is also achievable.

3.3 Two Examples

The capacity issue for multi-source network still remains open. To completely solve this problem, it seems that there is a long way to go. In this section, we use a method to estimate an outer bound for two networks. One is shown

to be tight. Another one has significant improvement from maxflow bound.

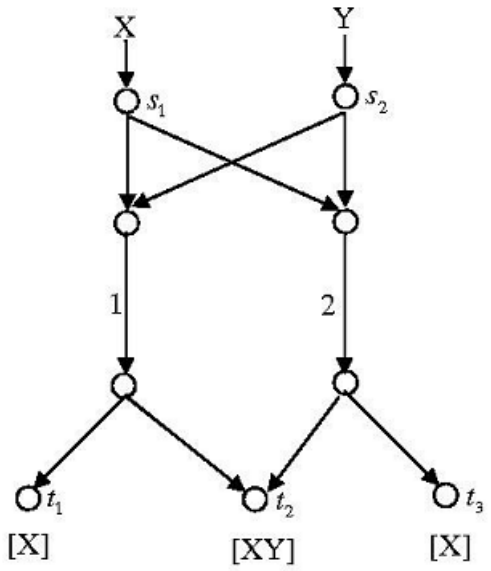

Figure 3.4: Network 1

First. let us look at the network depicted in Figure 3.4. Sink t_1 and t_2 demand source of X, while sink t_2 demand both information of X and Y. All edges have unit capacity. The maxflow bound for this network is

$$R = \{(\omega_1, \omega_2) : 0 \leq \omega_1 \leq 1, 0 \leq \omega_2 \leq 2, 0 \leq \omega_1 + \omega_2 \leq 2\}$$

The region is shown in Figure 3.5. Obviously, the extreme point $(1, 1)$ is not achievable, for if source node s_1 multicast one information bit of X to sink t_1 and t_2, then edge 1 and 2 are fully occupied by information of X, no information of Y can flow to sink t_2 from source node s_2. So the maxflow bound is not tight. Below we will develop an outer bound which is tight for this network.

Let U_1, U_2 represent the information flowing through edge 1 and 2. If sink

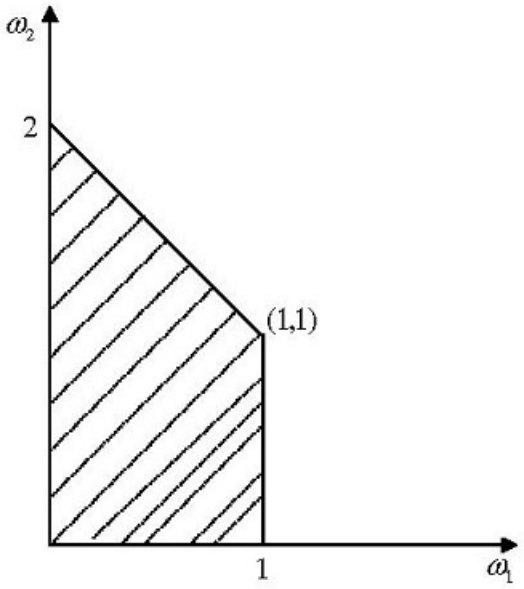

Figure 3.5: Maxflow Bound for Network 1

t_1 and t_2 can fully recover information of X, then U_1 and U_2 must contain full information of X separately, i.e.,

$$H(X|U_1) = H(X|U_2) = 0$$

From

$$\begin{aligned} H(X, U_1) &= H(U_1) + H(X|U_1) \\ &= H(X) + H(U_1|X) \end{aligned}$$

We have

$$H(U_1|X) = H(U_1) - H(X)$$

Similarly,

$$H(U_2|X) = H(U_2) - H(X)$$

Since X and Y are mutually independent,

$$\begin{aligned}
\omega_2 &= H(Y) \\
&= H(Y|X) \\
&\leq H(U_1, U_2|X) \\
&\leq H(U_1|X) + H(U_2|X) \\
&= H(U_1) + H(U_2) - 2H(X) \\
&\leq 2 - 2H(X)
\end{aligned}$$

the first Inequality is from the fact that edge 1 and 2 is a cut separating source node s_2 from t_2, so U_1 and U_2 must contain full information of Y. The third inequality if from the fact that the edge capacity if 1.

Therefore, we gain a new outer bound,

$$R^{'} = \{(\omega_1, \omega_2) : 0 \leq \omega_1 \leq 1, 0 \leq \omega_2 \leq 2, 2\omega_1 + \omega_2 \leq 2\}$$

As shown in Figure 3.6. It is easy to check this bound is tight. Let us look at another example in Figure 3.7. We also suppose each edge has unit capacity. The maxflow bound for this network is

$$R_0 = \{(\omega_1, \omega_2) : 0 \leq \omega_1 \leq 2, 0 \leq \omega_2 \leq 2, 0 \leq \omega_1 + \omega_2 \leq 3\}$$

But this bound is not tight, because the rate tuple (1,2) is not achievable. Before we give a new tighter bound, we prove a lemma,

Lemma 3.3.1 *X, U_1, U_2 are random variables defined on the same alphabet. If $H(X|U_1, U_2) = 0$, then $H(X|U_1) \leq H(U_2)$.*

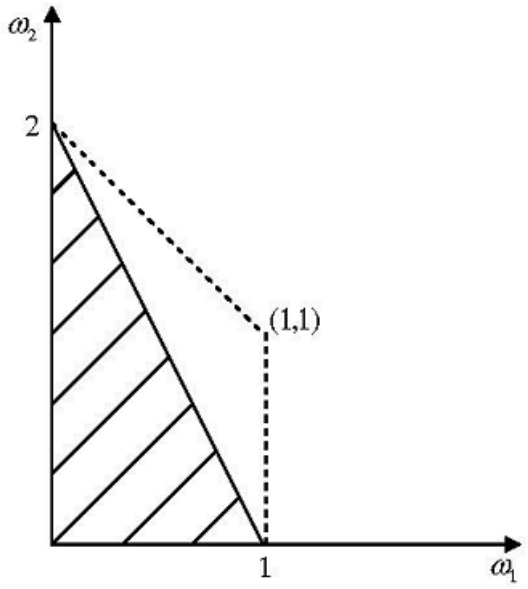

Figure 3.6: A Tight Bound for Network 1

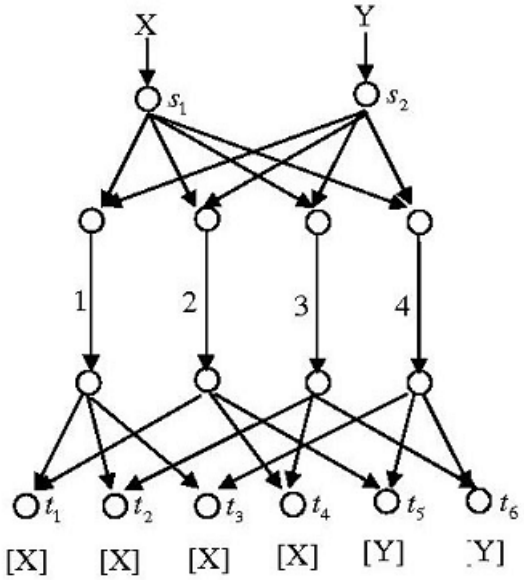

Figure 3.7: Network 2

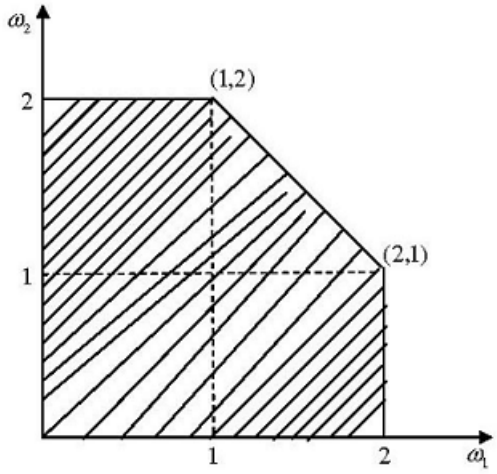

Figure 3.8: Maxflow Bound for Network 2

Proof:

$$\begin{aligned}
H(X) - H(X|U_1) &= I(X;U_1) \\
&= H(X) + H(U_1) - H(U_1, X) \\
&= H(X) + H(U_1) - (H(U_1, U_2, X) - H(U_2|U_1, X)) \\
&\geq H(X) + H(U_1) - H(U_1, U_2, X) \\
&= H(X) + H(U_1) - (H(U_2) + H(U_1|U_2) + H(X|U_1, U_2)) \\
&= H(X) - H(U_2) + I(U_1; U_2) \\
&\geq H(X) - H(U_2)
\end{aligned}$$

Hence, $H(X|U_1) \leq H(U_2)$. ∎

Now we turn to our problem. Sink t_4 demands information of Source X, and edge 2 and 3 is a cut separating s_1 from t_4. Let U_i denote the information

on edge i. Then we have

$$\begin{aligned} H(X) &= H(X|Y) \\ &\leq H(U_2, U_3|Y) \\ &\leq H(U_2|Y) + H(U_3|Y) \\ &= H(U_2, Y) - H(Y) + H(U_3, Y) - H(Y) \\ &= H(U_2) + H(U_3) + H(Y|U_2) + H(Y|U_3) - 2H(Y) \\ &\leq H(U_2) + H(U_3) + 2H(U_4) - 2H(Y) \\ &\leq 4 - 2H(Y) \end{aligned}$$

The third inequality is from the fact that $H(Y|U_2, U_4) = 0, H(Y|U_3, U_4) = 0$ and Lemma 3.3.1. So from sink t_4, we obtain an outer bound,

$$R_{\{t_4\}} = \{(\omega_1, \omega_2) : \omega_1 + 2\omega_2 \leq 4\}$$

Applying similar technique to sink t_6,

$$\begin{aligned} H(Y) &= H(Y|X) \\ &\leq H(U_3, U_4|X) \\ &\leq H(U_3|X) + H(U_4|X) \\ &= H(U_3, X) - H(X) + H(U_4, X) - H(X) \\ &= H(U_3) + H(U_4) + H(X|U_3) + H(X|U_4) - 2H(X) \\ &\leq H(U_3) + H(U_4) + H(U_1) + H(U_2) - 2H(Y) \\ &\leq 4 - 2H(X) \end{aligned}$$

So, we have

$$R_{\{t_6\}} = \{(\omega_1, \omega_2) : 2\omega_1 + \omega_2 \leq 4\}$$

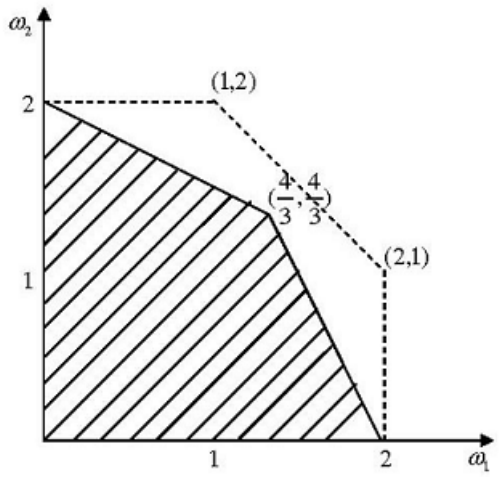

Figure 3.9: A New Outer Bound for Network 2

Furthermore, we can regard any $T \in T_X$ (or $\in T_Y$) as a node, and then apply the estimation technique above to obtain a bound for T. For example, we consider $\{t_1, t_2, t_4\}$,

$$\begin{aligned} 3H(X) &\leq H(U_1, U_2|Y) + H(U_1, U_3|Y) + H(U_2, U_3|Y) \\ &\leq 6 + 2H(Y|U_1) + 2H(Y|U_2) + 2H(Y|U_3) - 6H(Y) \\ &\leq 6 + 2H(Y) + 2H(U_4) + 2H(U_4) - 6H(Y) \\ &\leq 10 - 4H(Y) \end{aligned}$$

So,

$$R_{\{t_1,t_2,t_4\}} = \{(\omega_1, \omega_2) : 3\omega_1 + 4\omega_2 \leq 10\}$$

We obtain a new outer bound

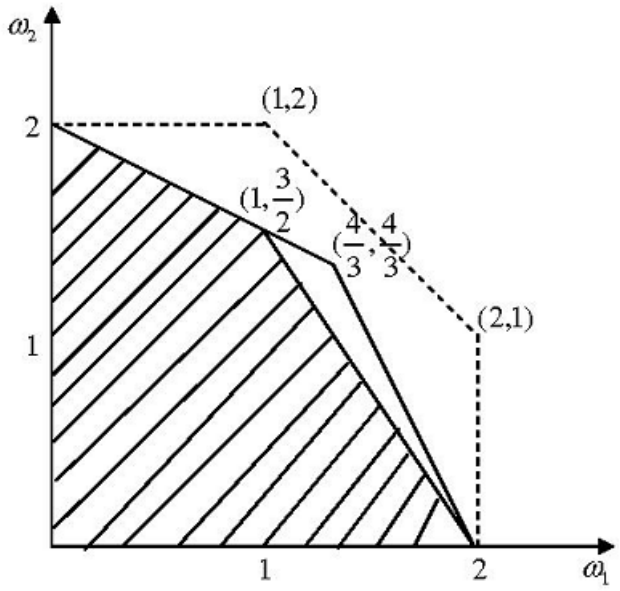

Figure 3.10: A Inner Bound for Network 2

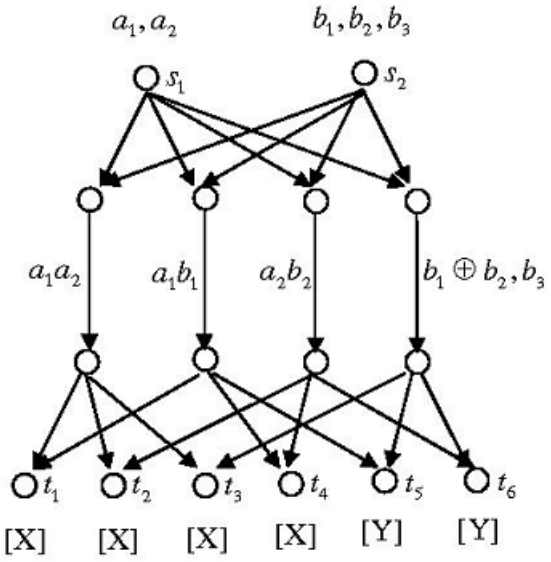

Figure 3.11: A Code That Achieves Rate $(\frac{3}{2}, 1)$

$$\tilde{R} = \{(\omega_1, \omega_2) : 2\omega_1 + \omega_2 \leq 4, \omega_1 + 2\omega_2 \leq 4\}$$

As Figure 3.9 show, the bound has significant improvement compared with maxflow bound.

Though we can not verify that whether rate tuple $(\frac{4}{3}, \frac{4}{3})$ is achievable or not. We can obtain an inner bound for Network 2, as shown in Figure 3.10. We further design a linear Boolean network code to achieve the rate tuple $(\frac{3}{2}, 1)$, this code is depicted in Figure 3.11.

3.4 Discussion

From last section, we see that the capacity issue for multi-source networks is highly non-trivial. In [13] and [14], an theoretical inner bound and outer bound are provided. But no numerical method is given to evaluate these bounds. Our method is straightforward and reveals some truth in the evaluation of achievable rate regions. In a cut, there may hide several copies of information. Sometimes we need to expand a cut to see how much information it can contain. Can the method in last two examples be generalized to general networks with two sources? It is extremely difficult to provide a universal close form for the outer bound of multi-source network coding. A more practical way is to develop an algorithm to evaluate the outer bound case by case.

Chapter 4

Conclusion and Future Work

4.1 Conclusion

In the thesis, we have designed a class of linear Boolean network codes with minimum network uses for combination networks. Using knowledge of linear algebra, we also prove that the codes constructed are optimal for $\binom{m}{2}$ and $\binom{m}{3}$ combination networks. Compared with the symbol-level linear codes constructed in [3] and [4] , linear Boolean network codes provide more insight into the nature of network codes.

During the evaluation of capacity for the two-source networks in Chapter 3, we have developed a straightforward method to estimate an outer bound for the networks. We also show that the outer bound is tight for some networks by designing linear Boolean network codes. Even with the simplified networks shown in our two examples, estimation of an outer bound is not so trivial. A lot of Shannon-type inequalities are involved. Hopefully, our method can shed some light on the multi-source network coding problem.

4.2 Future Work

Network coding has brought innovation to communication networks today. In the future, there are at least two directions to do research in network coding, one is theoretical, another one is related to application

The theoretical research on network coding is highly nontrivial. One problem is optimization problem. Given a network, how to design a network code with minimum network uses, and how to decrease the encoding and decoding complexity? Another problem need to be addressed is the multi-source problem. This problem is not completely solved for acyclic networks. And for cyclic networks, there is no result obtained so far. The performance of linear codes on multi-source networks is also deserved to investigate. What is the maximal rate for linear codes in multi-source networks? These problems are all yet to be solved.

Another direction for research on network coding is the application of network coding. With the mature of theoretical work for single source networks, many researchers direct their research into applications of network coding to the computer networks and wireless ad hoc networks. Lots of problems arise from applications of network coding. One important problem is the synchronization problem for real time applications. We need to combine network coding with other communication techniques to address these problems.

Bibliography

[1] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network Information Flow," *IEEE Trans. Inform. Theory*, vol. 46, no. 4, pp. 1204–1216, Jul. 2000.

[2] L. K. Ford and D. K. Fulkerson, *flows in networks*. Princeton, N. J: Princeton Univ. Press, 1962.

[3] S. Jaggi, P. Sanders, P. A. Chou, M. Effros, S. Egner, K. Jain, L. Tolhuizen, "Polynomial Time Algorithms for Multicast Network Code Construction," *IEEE Trans. Inform. Theory*. vol. 51, no. 6, pp. 1973–1982, June 2005.

[4] S.-Y. R. Li, R. W. Yeung, and N. Cai, "Linear Network Coding," *IEEE Trans. Inform. Theory*, vol. 49, no. 2, pp. 371–381, Feb. 2003.

[5] X. B. Liang, "Matrix games in the multicast networks: Maximum information flows with network switching," *IEEE Trans.Inform.Theory*, to be published, June 2006.

[6] R. W. Yeung, S.-Y. R. Li, N. Cai, and Z. Zhang, "Theory of Network Coding," submitted to *Foundations and Trends in Communications and Information Theory*, preprint, 2005.

[7] F. J. McWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*. Amsterdam, The Netherlands: North-Holland, 1977.

[8] T. Ho, R. Koetter, M. Médard, D. Karger, and M. Effros, "The benefits of coding over routing in a randomized setting," in *Proc.IEEE Int. Symp.Information Theory (ISIT)*, Yokohama, Japan, June /July 2003, pp. 442.

[9] R. Koetter and M. Médard, "An Algebraic Approach to Network Coding," *IEEE/ACM Transactions on Networking*, vol. 11, no. 5, pp.782–795, Oct. 2003.

[10] S. Riis, "Linear Versus Non-linear Boolean Functions in Network Flow," *Proceeding of CISS* 2004.

[11] R. Dougherty, C. Freiling and K. Zeger, "Linearity and Solvability in Multicast Networks," *IEEE Trans.Inform.Theory*, vol. 50, no. 10, pp. 2243–2256, Oct. 2004.

[12] R. Dougherty, C. Freiling and K. Zeger, "Insufficiency of Linear Coding in Network Information Flow," *IEEE Trans.Inform.Theory*, vol. 51, no. 8, pp. 2745–2759, Aug. 2005.

[13] L. Song and R. W. Yeung, "Zero-error Network Coding for Acyclic Networks," *IEEE Trans. Inform. Theory*, vol. 49, no. 12, pp. 3129–3139, Dec. 2003.

[14] R. W. Yeung, *A First Course in Information Theory*. New York: Kluwer/Plenum, 2002.

[15] X. Yan, J. Yang and Z. Zhang, "An outer Bound for Multi-source Multisink Network Coding with Minimum Cost Consideration," submitted to *IEEE Trans. Inform. Theory*, preprint, 2005.

[16] E. Erez and M. Feder, "Capacity region and network coding for two recievers multicast with private and common data," *Workshop on Coding, Cryptography and Combinatorics*, 2003.

[17] C. K. Ngai and R. W. Yeung, "Multisource network coding with two sinks," in *International Conference on Communications, Circuits and Systems (ICCCAS)*, June 2004.

[18] P. A. Chou, Y. Wu and K. Jain, "Practical Network Coding," in *Allerton Conference on Communication, Control and Computing*, Monticello, IL, 2003.

[19] C. Gkantsidis and P. R. Rodriguez, "Network Coding for Large Scale Content Distribution," in *IEEE Infocom*, 2005.

[20] http://www.mit.edu/%7Emedard/coding1.htm.

[21] C. K. Ngai and R. W. Yeung, "Network Coding Gain of Combination Networks," in *IEEE Information Theory Workshop*, San Antonio, pp. 283–287, Oct. 2004.

Vita

Shoupei Li was born in Guangdong, China, on October 1, 1978. After finishing his high school in Guangdong, he went to Shanghai, where he receives Bachelor of Science and Master of Science degrees both in mathematics from Fudan University, Shanghai, China. In August, 2004, he came to Louisiana State University to pursue graduate studies in electrical and computer engineering with specialization in communications and digital signal processing. He is currently a candidate for Master of Science in electrical engineering degree, which will be awarded in May 2006.